Object-Oriented Design

Object-Oriented Design

- We will cover the following topics
  - What object-oriented means
  - The difference between object-oriented design and object-oriented programming
  - The basic principles of object-oriented design
  - Basic Unified Modeling Language (UML)

Object-Oriented

- An object is normally a tangible thing that we can sense, feel, and manipulate like baby toys, wooden blocks, plastic shape, etc.
- The definition of an object in software development: Objects may not be tangible things that you can pick up, sense, or feel, but they are models of something that can do certain things and have certain things done to them.
    - **A software object is a collection of data and associated behaviors.**
- Object-oriented is a technique used for modeling complex systems by describing a collection of interacting objects via their data and behavior.
- Software development typically has 3 stages:
    - **Object-oriented analysis (OOA)** is the process of looking at a problem and identifying the objects and interactions between those objects.
        - The analysis stage is all about what needs to be done.
        - The output of the analysis stage is a set of requirements.
    - **Object-oriented design (OOD)** is the process of converting such requirements into an implementation specification.
        - The design stage is all about how things should be done.
        - The output of the design stage is an implementation specification.
    - **Object-oriented programming (OOP)** is the process of converting this perfectly-defined design into a working program.
- **Iterative development mode**l where a small part of the task is modeled, designed, and programmed, and then the program is reviewed and expanded to improve each feature and include new features in a series of short development cycles.

Reference: Python 3 Object-Oriented Programming - Third Edition By Dusty Phillips

Objects

- You can see the world is a collection of objects. The things you have in your house are objects. Things your throw away are objects, and things you want to buy are objects..
- An object can be a person, place, thing, concept, or possibly event.

Attributes and Behaviors

- A person as well as a house, car, and any other real-world objects are described by using two groups of features: attributes and behaviors.
  - **Attribute**: is a characteristic of an object.
    - Example, a person has a first name, last name, height, and weight.
  - **Behavior** :is an action that an object is capable of performing.
    - Example, a person sits, stands, walks, and runs, among thousands of other behaviors a person performs.

	Attributes	**Behaviors**
automobile	width, height, weight, wheels, an engine	move in a direction, stop, is steered to a different direction
sales order form	customer name, customer address, item ordered, amount due	collecting information, modifying information, and processing the sales order

Exercises- Attributes and Behaviors

- Complete the following table

	Attributes	Behaviors
Apple		
Book		
Computer		
Color		
Chair		
Table		
Rectangle		
Circle		
Triangle		

Why do we use objects?

By focusing on objects make it easy for us to understand complex problems. Objects enable us to look at details that are of interest to us and ignore other details.

Inheritance

It is a way for one object to receive attributes and behaviors of another object. This is called a "is a" relationship. Example, the Person and Student objects:

Person object

Attributes	Behaviors
First name	Sitting
Last name	Standing
Address	Walking
Telephone	Running

In addition, Student has attributes and behaviors that are unique to a student

Attributes	Behaviors
Student ID	Taking a test
Program	Attending a class
Grade point average	Doing homework

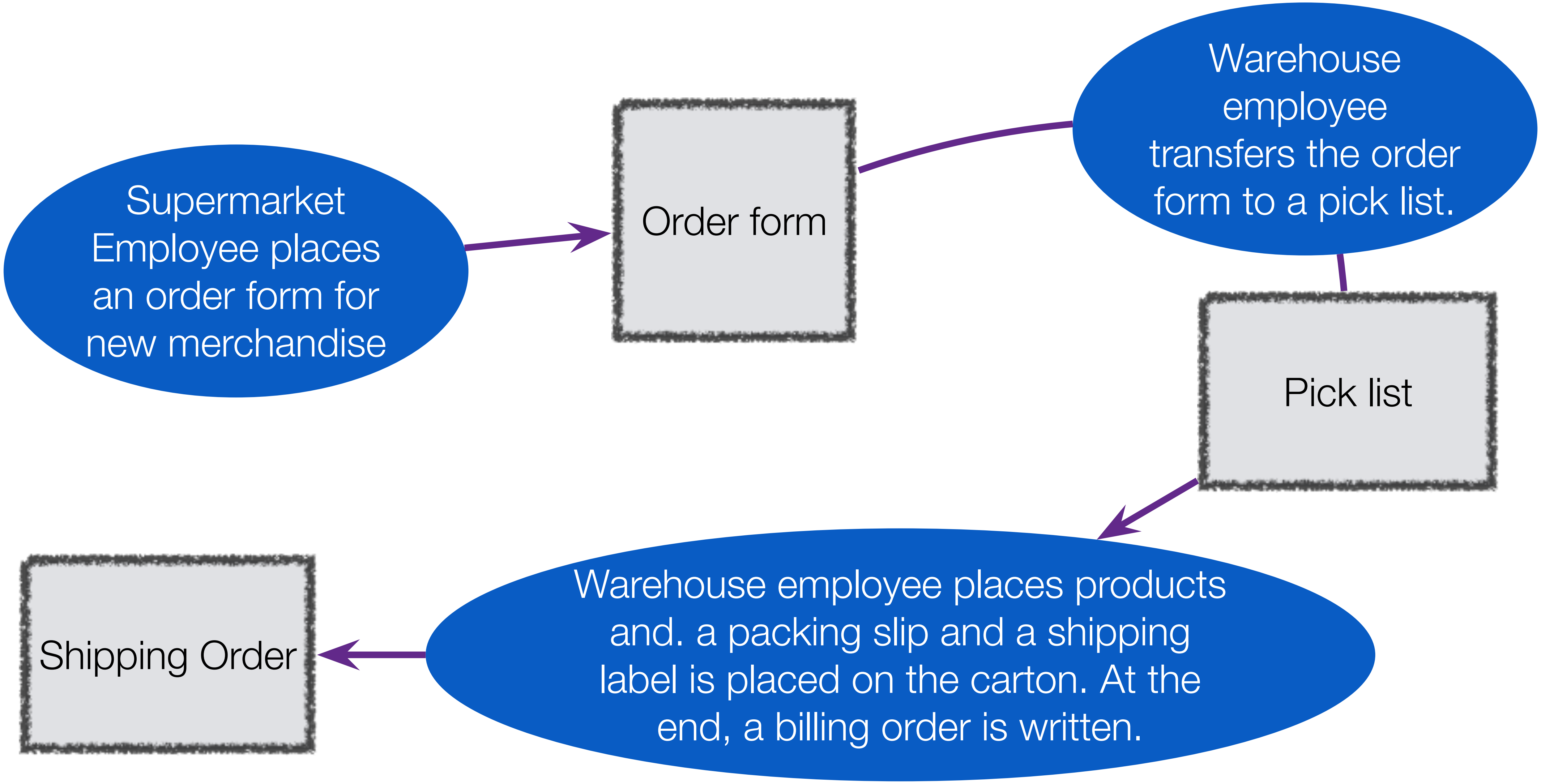
Inheritance

It makes sense that Student inherits attributes and behaviors of Person because Student has an "is a" relationship with Person. This means we creates only an instance of Student can access attributes and behaviors of Person without creating an instance of Person explicitly.

Attributes	Behaviors
First name	Sitting
Last name	Standing
Address	Walking
Telephone	Running
Student ID	Taking a test
Program	Attending a class
Grade point average	Doing homework

Case Study - Supermarket

Let’s take a look at a supermarket to get a better understanding of a business system and the objects used in business. Each week supermarket employees survey the shelves and the storeroom to determine what products to order from the central warehouse.



Supermarket Employee places an order form for new merchandise
Order form
Warehouse employee transfers the order form to a pick list.
Pick list
Warehouse employee places products and. a packing slip and a shipping label is placed on the carton. At the end, a billing order is written.
Shipping Order

Case Study - Supermarket

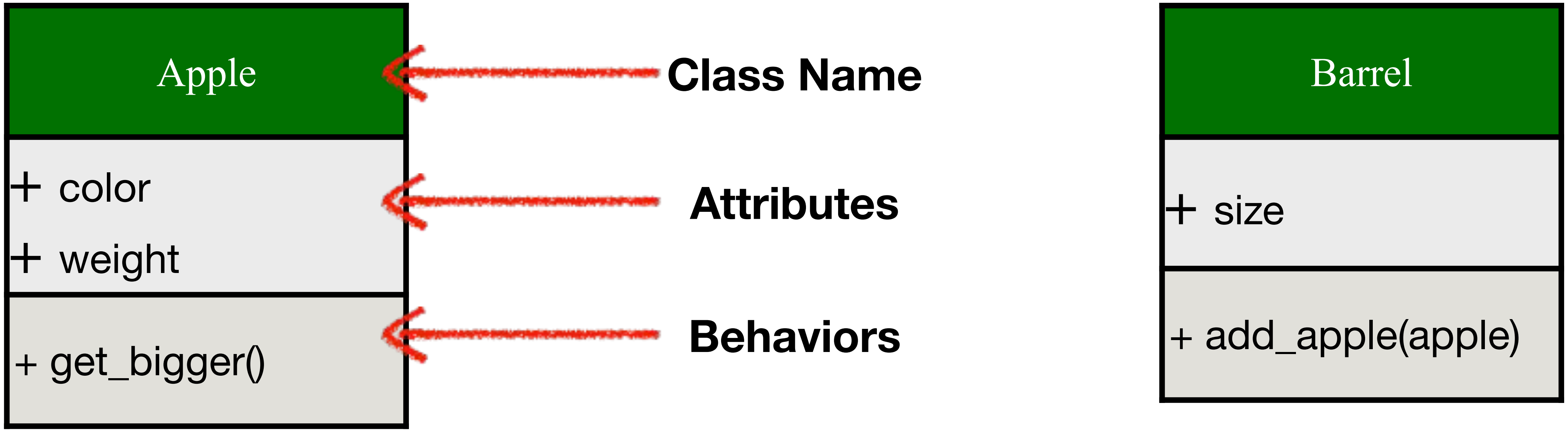
- There are many objects in the system a supermarket uses to order merchandise to restock shelves. These include an employee, a building, cartons, and a truck. However, we are interested in objects of the system, such as the form used to order merchandise
- What business objects do you see in the supermarket ordering system?
    - Order form
    - Pick list
    - Packing slip
    - Shipping order
    - Shipping label
- It is hard to imagine the above objects having a behavior because it doesn’t do anything. It is not like a window that opens and closes.
- However, an object does have behaviors. Here are a few behaviors of an order form object:
    - Enter information
    - Modify information
    - Delete modified information
    - Display the form
    - Process the order
    - Modify the order
    - Cancel the order

Exercise - NPU Library

- In NPU library, there are many objects in the system a librarian uses to order right books and magazines to restock bookshelves, and also students to borrow books. These include a librarian, books, magazines, students, and faculties. For now, we are only interested in objects of the library system, such as computer forms or web pages used for students to search and borrow books.
- What business objects do you see in the NPU Library system?
- What are the behaviors of these objects?

UML

- The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
- UML Class Diagram is used to describe the classes of an object-oriented application.
  - The plus sign (+) marks the fields and methods that can be assessed by other classes.
  - Attributes - are frequently referred to as data members or properties
  - Behaviors - are frequently referred to as data functions or methods



Apple

+ color

+ weight

+ get_bigger()

Class Name

Attributes

Behaviors

Barrel

+ size

+ add_apple(apple)

Naming Conventions

- In computer programming, a naming convention is a set of rules for choosing the character sequence to be used for identifiers which denote variables, types, functions, and other entities in source code and documentation.

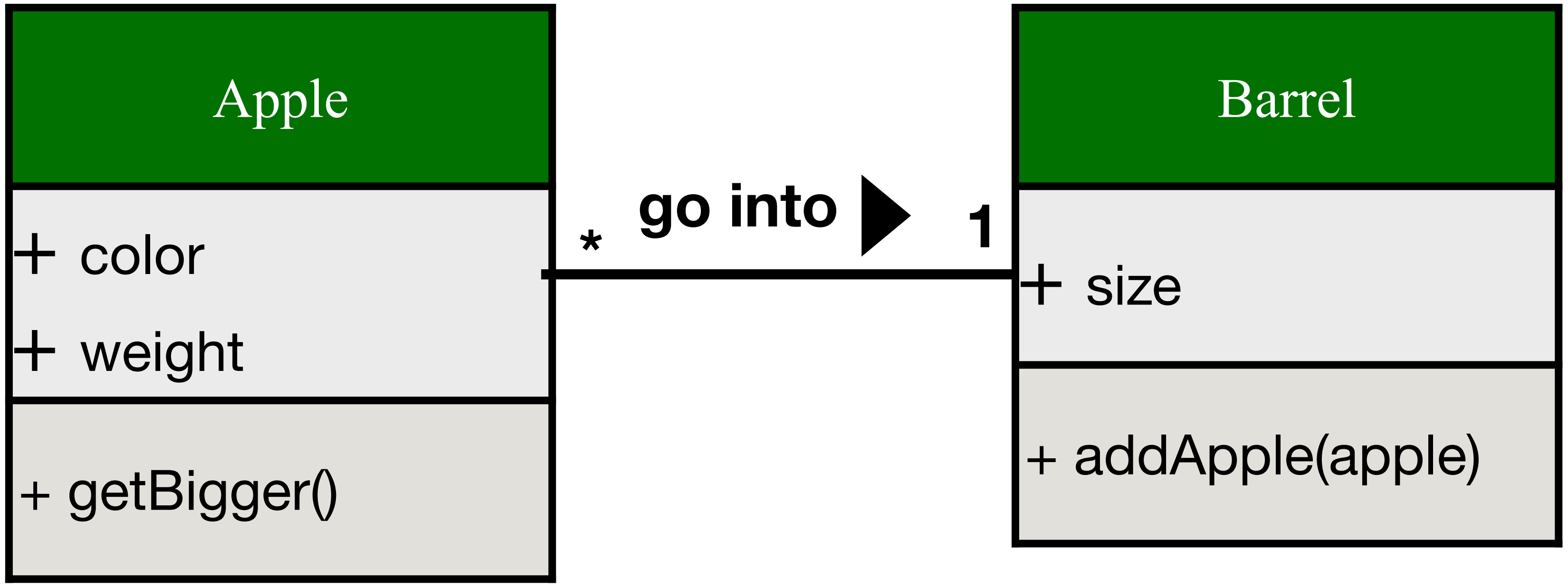
- For Python language, we can refer to the following link: PEP 8 -- Style Guide for Python Code

- Class Names - should normally use CapWords preferring short names: T, AnyStr, Num.

- Method Names and Instance Variables - use the function naming rules: lowercase with words separated by underscores as necessary to improve readability. E.g. my_book, the_table, get_name, etc

Class Relationship

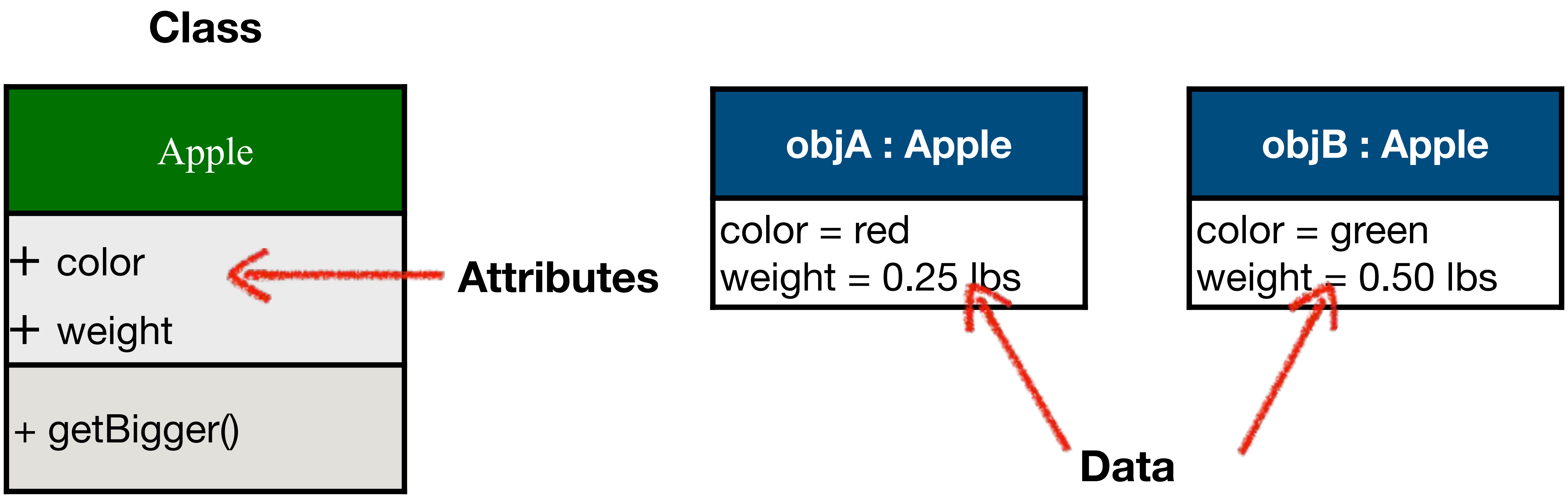
- There are two objects in the following picture: apple and barrel.
- As apples go in barrels in the picture, an Apple is also somehow associated with a Barrel.
- Association is the most basic way for two classes to be related.
- The association between Apple and Barrel is many apples go in one barrel.
- The following class diagrams tells us that apples go in barrel, with a little arrow showing what goes in what.
- One Barrel can hold many (represented by a *) Apple objects. Any one Apple can go in exactly one Barrel. This number is referred to as the **multiplicity** of the object.



Reference: Python 3 Object-Oriented Programming - Third Edition By Dusty Phillips

Data describes object

- An object instance has its own set of data and behaviors.
- Data represents the individual attributes of a certain object.
- Attributes are frequently referred to as members or properties



Class

Apple

+ color

+ weight

+ getBigger()

Attributes

objA : Apple

color = red
weight = 0.25 lbs

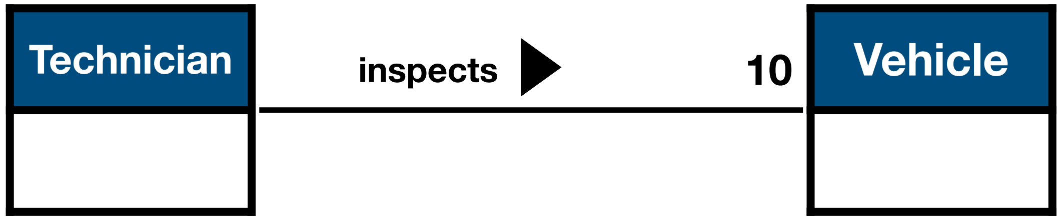
objB : Apple

color = green
weight = 0.50 lbs

Data

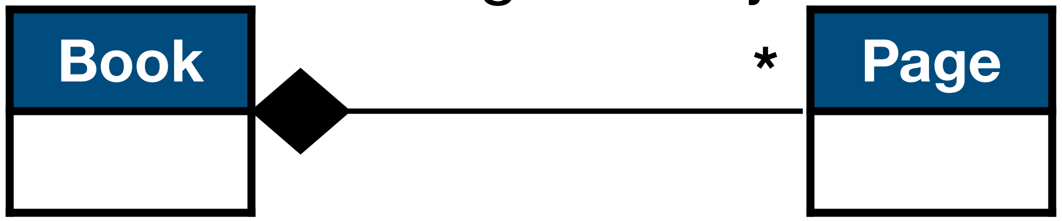
Association vs Aggregation vs Composition

- Association - if two classes need to communicate with each other, there must be a link between them, and that can be represented by an association.
  - Optionally, an arrow can be added to indicate the navigation direction.
  - Multiplicity is the number of instances of one class related to one instance of the other class.

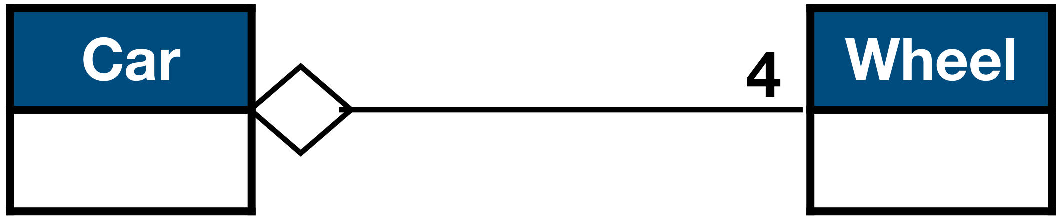


inspects is the name of the association

- Composition - is a part of an association relationship. The composition is considered as a strong type of association. Objects are highly dependent upon each other. Deleting one object affects another associated object.



- Aggregation - is a part of an association relationship. Aggregation is considered as a weak type of association. Objects are not dependent upon the other object. Deleting one object does not affect another associated object.

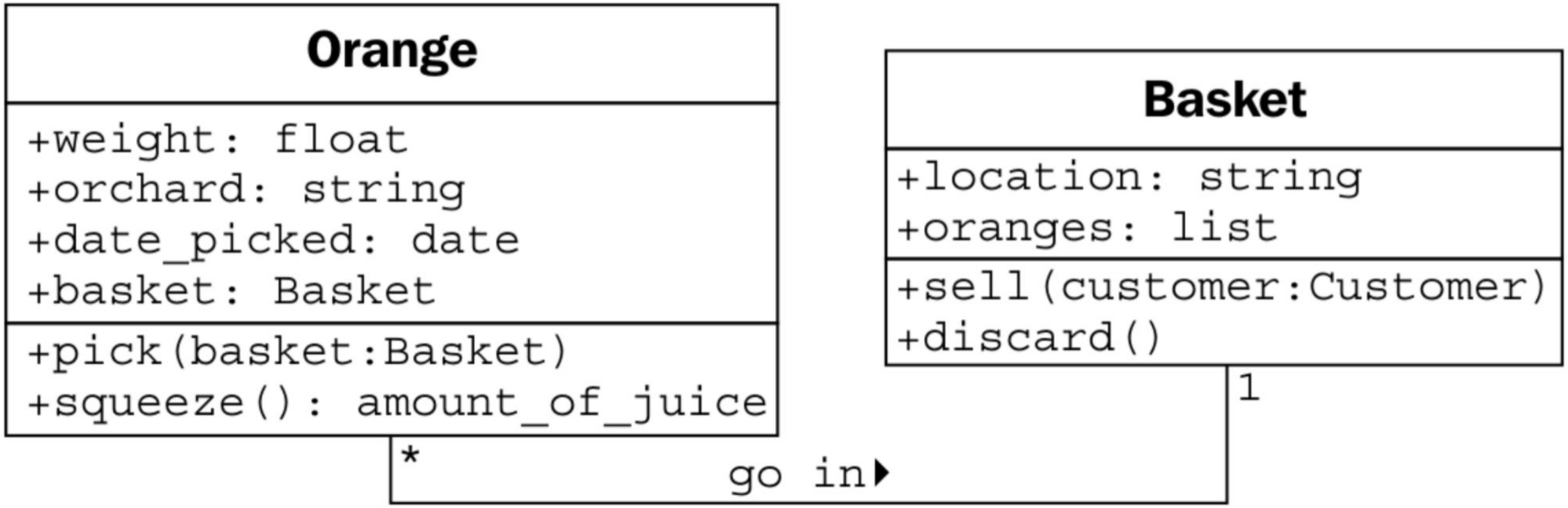


Exercises - Association, Composition and Aggregation

- Define and draw associations for the following classes
  - Teacher and Student
  - Person and Hand
  - Classroom and Student
  - File and Folder
  - Book and Bookshelf
  - House and Bathroom
  - Car and Engine

Behaviors are actions

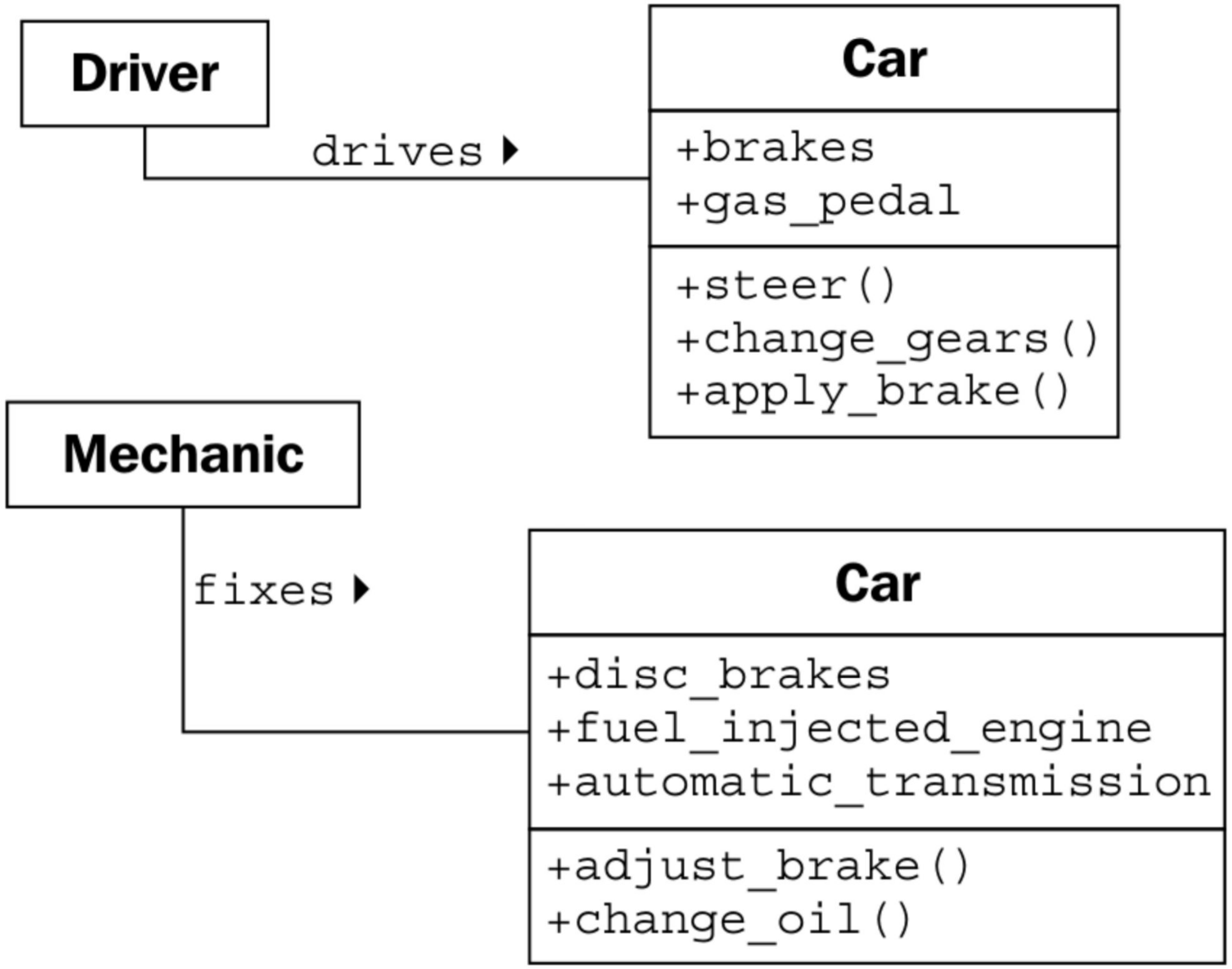
- Behaviors are actions that can occur on an object.
- The behaviors that can be performed on a specific class of object are called **methods**
- **Orange** class has two actions:
  - pick(basket) - Place the orange in a basket
  - squeeze() - return the amount of juice retrieved
- Basket class has two actions:
  - sell(customer) - Sell the basket to a specific customer
  - discard() - Discard the basket of bad oranges.



Reference: Python 3 Object-Oriented Programming - Third Edition By Dusty Phillips

Hiding details and creating the public interface

- The key purpose of modeling an object in object-oriented design is to determine what the public interface of that object will be.
- The interface is the collection of attributes and methods that other objects can access to interact with that object.
- This process of hiding the implementation of an object is suitably called information hiding (encapsulation).
- Abstraction is another object-oriented term related to encapsulation and information hiding.
- Abstraction means dealing with the level of detail that is most appropriate to a given task.



Reference: Python 3 Object-Oriented Programming - Third Edition By Dusty Phillips

What Software You Need

- Download and Install the latest version of Python Distribution that includes IDLE for Python application development.
  - https://www.python.org/downloads/
- Optional software
  - You can install Gliffy Diagrams Chrome Extension for drawing UML Diagrams.
    - https://chrome.google.com/webstore/detail/gliffy-diagrams/bhmicilclplefnflapjmnngmkkkkpfad?hl=en
  - You can also install Visual Studio Code for more advanced code editor.
    - Download Visual Studio Code
    - Python in Visual Studio Code